

## Governors State University OPUS Open Portal to University Scholarship

---

All Capstone Projects

Student Capstone Projects

---

Fall 2015

### GSU Alumni Portal

Vemuri Vinusha Chowdary  
*Governors State University*

Sairam Dasu  
*Governors State University*

Siva Machineni  
*Governors State University*

Follow this and additional works at: <http://opus.govst.edu/capstones>

 Part of the [Computer and Systems Architecture Commons](#), and the [Computer Sciences Commons](#)

---

#### Recommended Citation

Chowdary, Vemuri Vinusha; Dasu, Sairam; and Machineni, Siva, "GSU Alumni Portal" (2015). *All Capstone Projects*. 163.  
<http://opus.govst.edu/capstones/163>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to  
[http://www.govst.edu/Academics/Degree\\_Programs\\_and\\_Certifications/](http://www.govst.edu/Academics/Degree_Programs_and_Certifications/)

Visit the [Governors State Computer Science Department](#)

This Project Summary is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact [opus@govst.edu](mailto:opus@govst.edu).

# Table of Contents

<b>1</b>	<b><i>Project Description</i></b> .....	1
1.1	Project Abstract .....	1
1.2	Competitive Information .....	1
1.3	Relationship to Other Applications/Projects .....	1
1.4	Future Enhancements .....	1
1.5	Definitions and Acronyms .....	1
<b>2</b>	<b><i>Technical Description</i></b> .....	2
2.1	Project/Application Architecture.....	2
2.2	Project/Application Information flows.....	3
2.3	Capabilities .....	4
2.4	Risk Assessment and Management .....	4
<b>3</b>	<b><i>Project Requirements</i></b> .....	4
3.1	Identification of Requirements .....	4
<b>4</b>	<b><i>Project Design Description</i></b> .....	5
<b>5</b>	<b><i>Project Design Units Impacts</i></b> .....	17
5.1	Functional Area/Design Unit A.....	22
5.1.1	Functional Overview .....	22
<b>6</b>	<b><i>Conclusion</i></b> .....	29
<b>7</b>	<b><i>Acknowledgements</i></b> .....	30
<b>8</b>	<b><i>References</i></b> .....	31
<b>9</b>	<b><i>Appendix</i></b> .....	32

## ***1 Project Description***

### ***1.1 Project Abstract***

The main objective of Gsu alumni portal web application is to allow old and new students of a university to communicate with each other. The application allows students to register and then search the data based on different criteria. Also it has the benefit of having a centralized database and up to date information. A user can easily obtain information about other registered users. This application is deployed using Cloud computing (Microsoft Azure) and nothing is managed locally. The administrator is responsible for maintaining information of students. When a student submits the registration form, administrator will complete the verification process and, if successful, the student details are added into the database. The administrator maintains the passwords of Event Manager and that of himself. Event Manager Module maintains the information about various events organized by the university. Details of notifications are also maintained. Event manager can add, edit and view event details. And event manager sends an SMS about the events to the registered students and Administrator. The Students after successful registration process, they can logon into their account and can send mails, post queries, update their profiles and even search for other student details. One can view the event details and search for specific information. The module provides mail and query functionalities.

### ***1.2 Competitive Information***

The GSU Alumni portal web application is deployed using Cloud computing, Microsoft azure as a platform. We can overcome single point Hardware failure by using cloud computing. The other pros of cloud computing are no hardware to maintain, unlimited instance scaling, unlimited disk scaling, Dynamic scaling, pay for what we use, Resilient and Redundant.

### ***1.3 Relationship to Other Applications/Projects***

The GSU Alumni portal Db is related to MSSQL, Views are related to visual studio using Asp.net technology. The local Db is connected to SQL Azure (it can be managed in cloud too). The local Asp.net project is published in cloud using windows azure as a platform.

### ***1.4 Future Enhancements***

For Future Enhancements the GSU alumni portal can have an automated e-mail, when a new event gets updated. Furthermore admin can have an automated e-mail, whenever a new student registers in the Portal instead of constantly checking the student Registrations.

### ***1.5 Definitions and Acronyms***

- SSMS - Sql Server Management Studio.
- SQL - Structured Query Language.
- SDK - Software Development Kit.
- SMTP - Simple Mail Protocol.

## 2 Technical Description

### 2.1 Project/Application Architecture

- While Developing the GSU Alumni Portal web application the below mentioned architecture is implemented using Asp.net Technology.

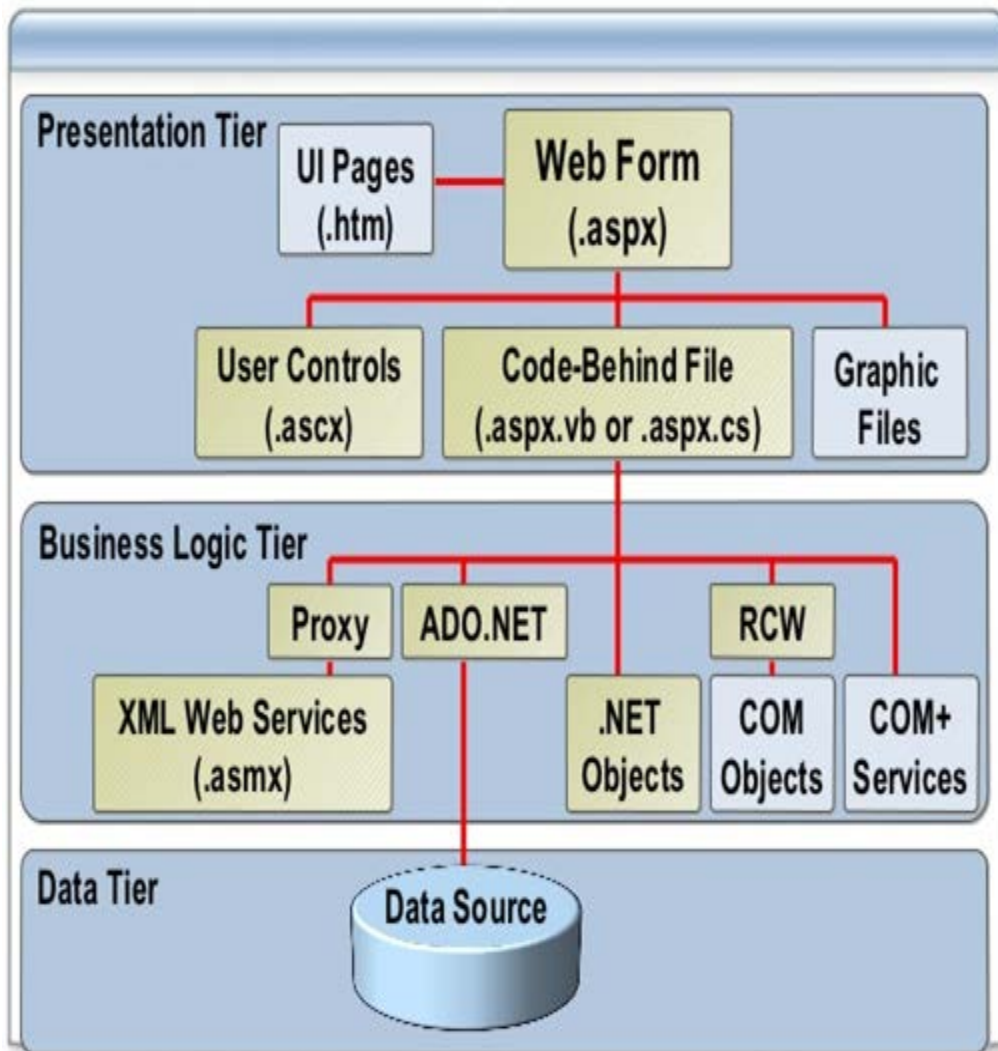


Figure 1(Asp.net web application architecture)

- The asp.net web application is deployed using Microsoft windows azure. After the completion of the business logic using Asp.net, the application is deployed by using the following architecture.

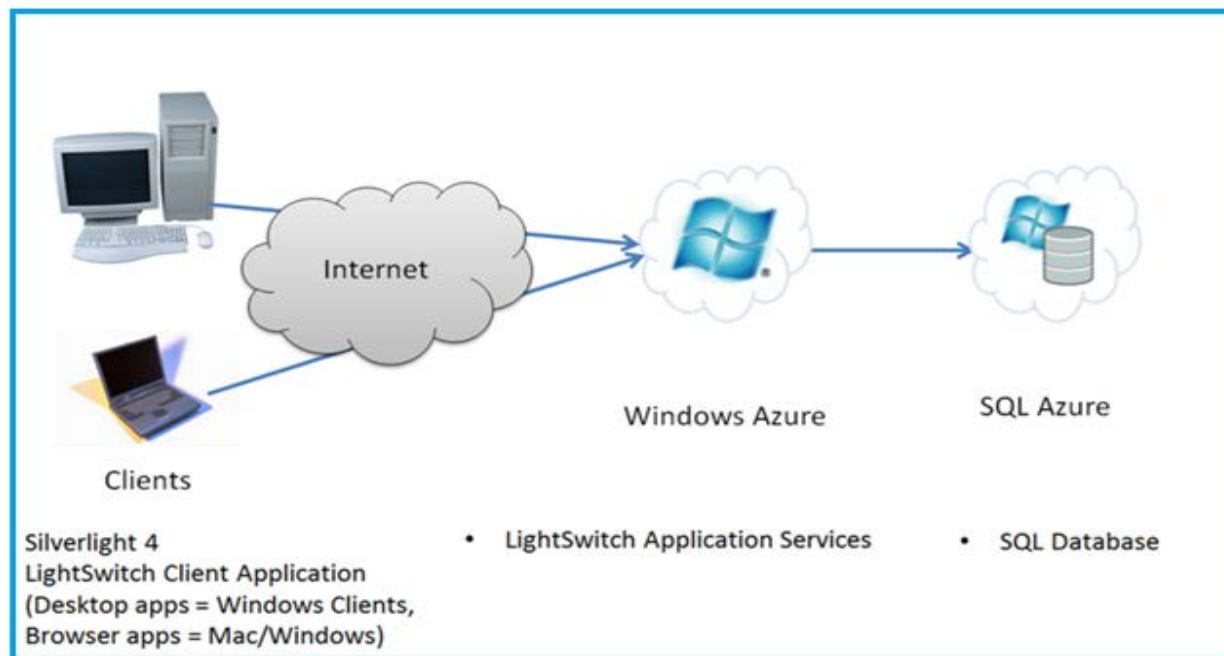


Figure 2(3- Tier Azure Deployment)

## 2.2 Project/Application Information flows

The application information flows through the Administrator, Event Manager and students Modules.

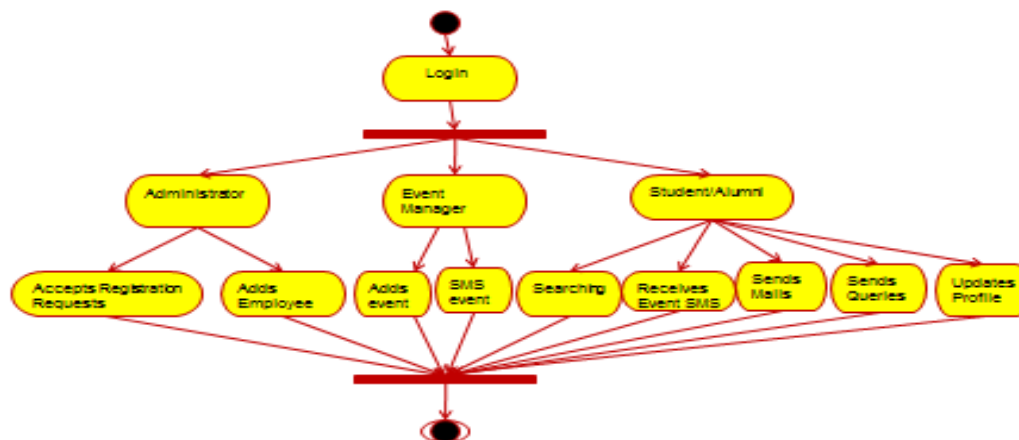


Figure 3 (Sequence Diagram)

## **2.3 Capabilities**

The Database application provides capabilities to support business application such as retrieving/adding/deleting/updating user data. The connection string should be placed appropriately between database layer and Asp.net web application while developing the Project.

## **2.4 Risk Assessment and Management**

The risks involved in this project are unsupportive versions of software, while publishing the web application to cloud using Microsoft windows azure. The risks can be managed by choosing the compatible software requirements.

## **3 Project Requirements**

### **3.1 Identification of Requirements**

This Project is mainly divided in to three modules Administrator, Event Manager and student. To implement these modules successfully the following project requirements are maintained in this project.

- Software Requirements:
  - 1) Visual studio 2015.
  - 2) SQL Server Management studio 2015.
  - 3) Microsoft .Net Frame work 4.0(Minimum).
  - 4) Azure SDK.
  - 5) Microsoft Web Platform Installer 4.6.
- Hardware Requirements:
  - 1) RAM: 8GB.
  - 2) Available Hard Disk Space: 40GB.
- Web hosting Requirement:
  - 1) Microsoft windows pay-as-you-go account.

#### 4 Project Design Description

- Database Design

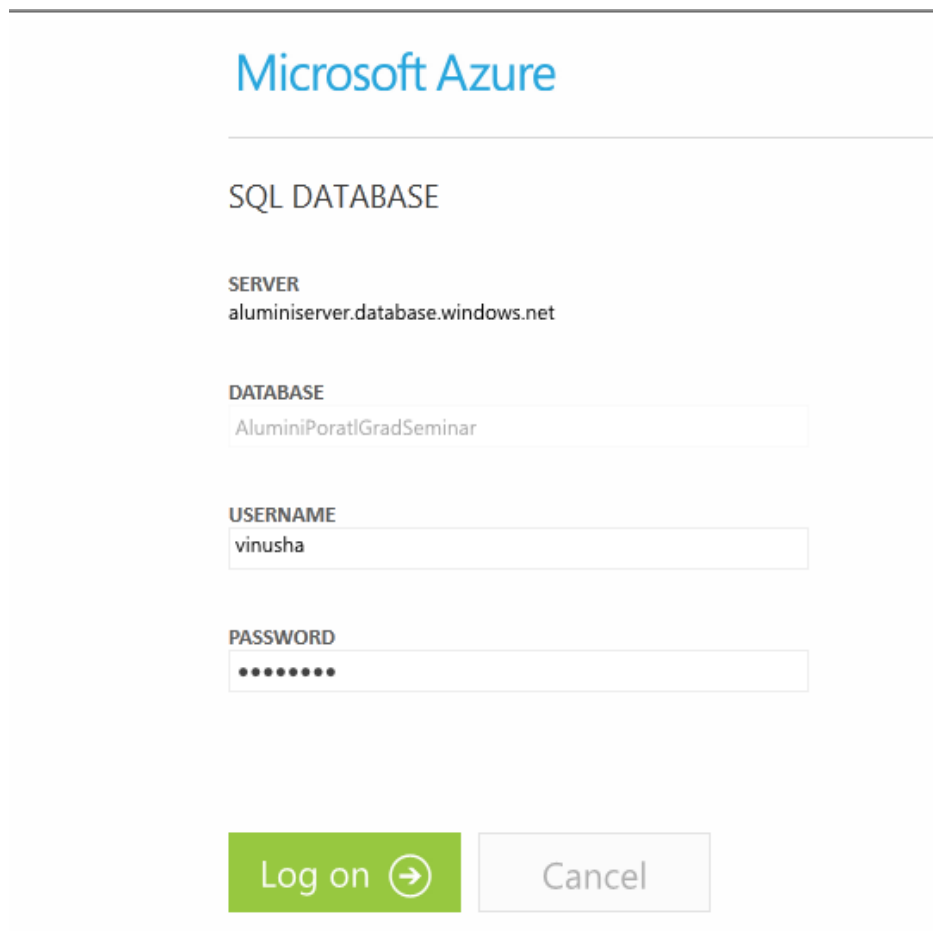
The project is designed using asp.net as front end and sql server database as backend. The sql database is managed in the cloud by using windows azure sql Databases. The database is designed using Tables and stored procedures. The database can be managed in the cloud by using the following URL.

URL: [https://aluminiserver.database.windows.net/?langid=en-us#\\$database=AluminiPoratIGradSeminar](https://aluminiserver.database.windows.net/?langid=en-us#$database=AluminiPoratIGradSeminar)

This url directs to full functionalities of the database, where the database can be managed in cloud with the valid server credentials. As discussed the database the project is designed using Tables and stored procedures. The complete view of database design is provided in the below screen shots. The database design is explained using Dependency diagrams.

- Database Login:

---



Microsoft Azure

SQL DATABASE

SERVER  
aluminiserver.database.windows.net

DATABASE  
AluminiPoratIGradSeminar

USERNAME  
vinusha

PASSWORD  
••••••••

Log on → Cancel

Figure 4 (Azure Sql Database Login)

After log on in to Sql database server, we can view the existing data, queried and database data can be changed according to the requirement.

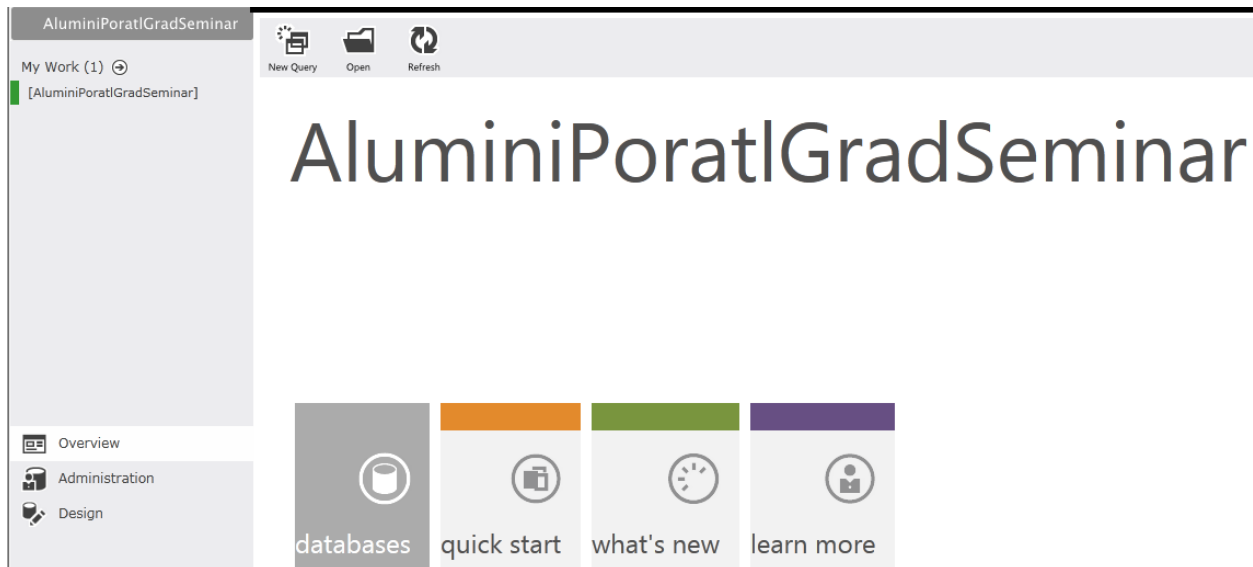


Figure 5(Azure Sql Database name)

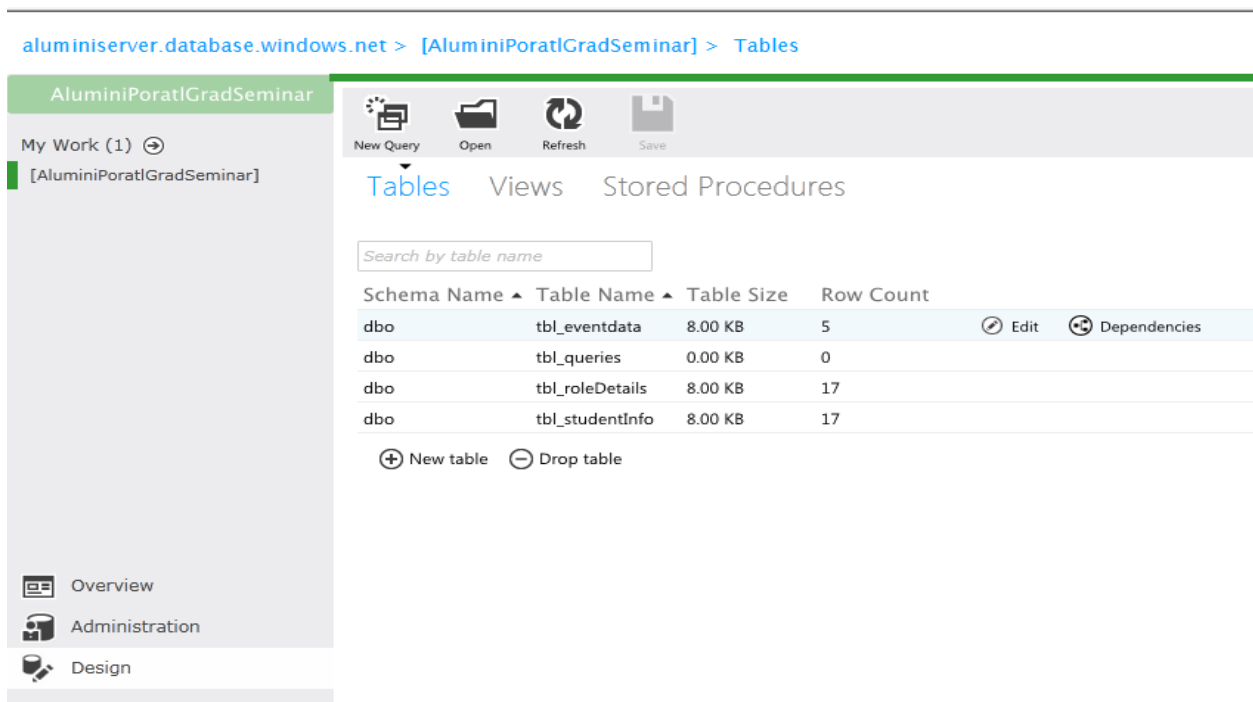


Figure 6 (Database Parameters – List of Tables)



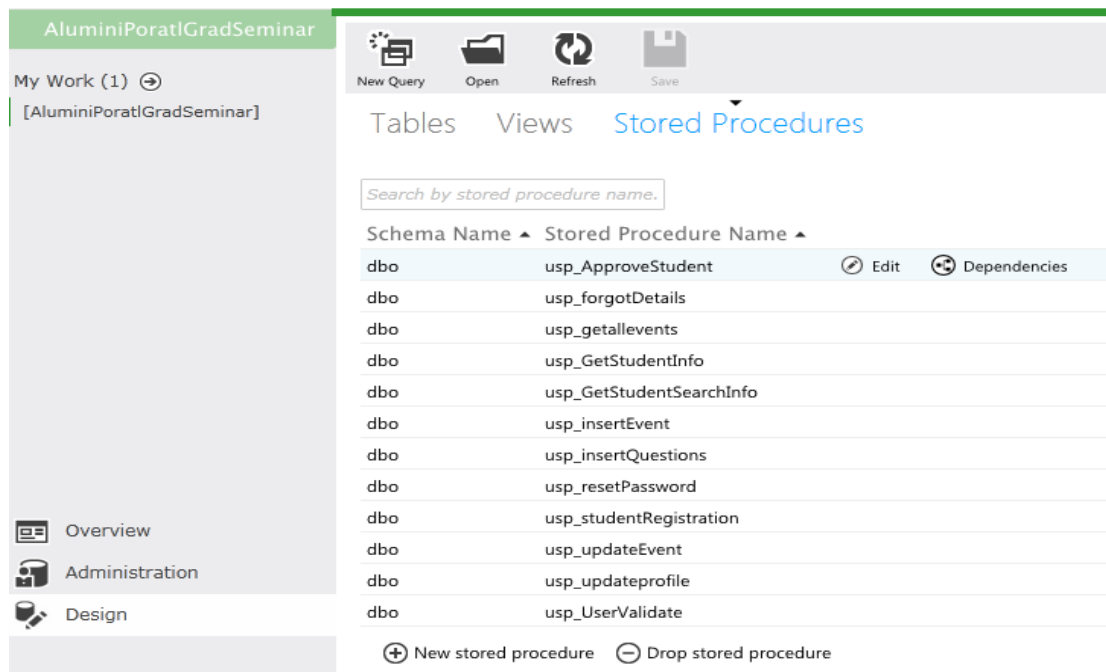
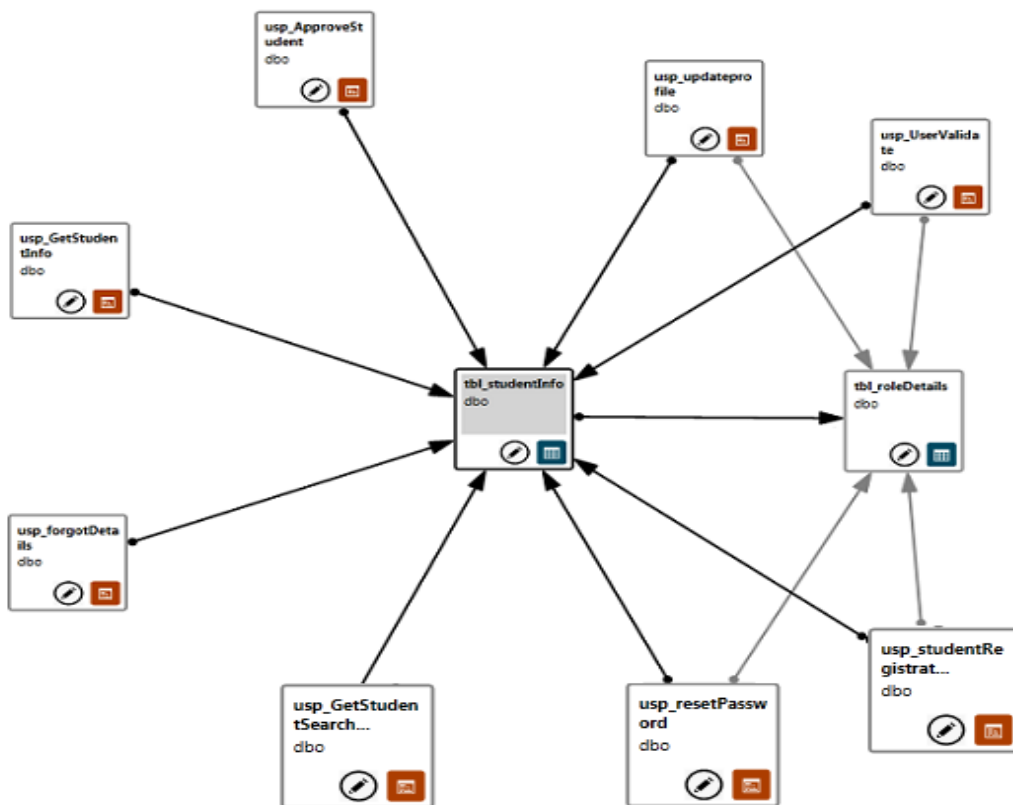


Figure 7(List of Stored Procedures)

This figure illustrates the all the dependencies.



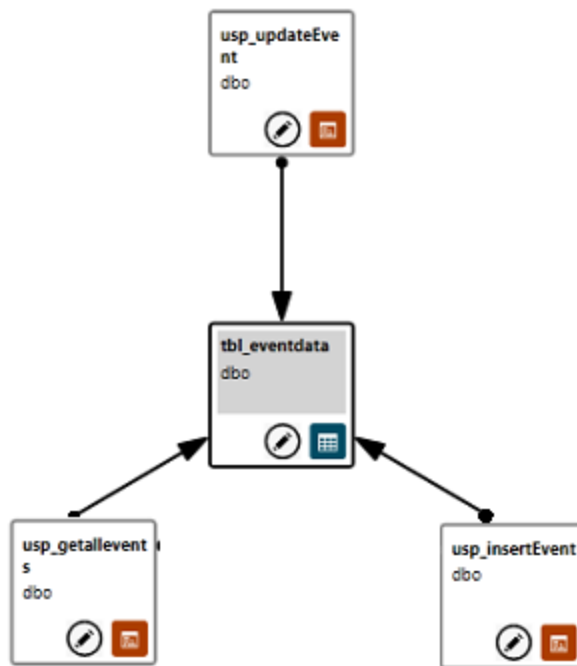


Figure 8(Dependencies)

- Front End Design:

The front end pages are designed using Asp.net. The code behind the business logic is discussed with the design view below. The main code functionalities are in Home, Dashboard, Event dashboard and Student dashboard.

Home.aspx: The code in Home.aspx page mainly deals with login credentials for administrator, event manger and student. The student registrations are also included in home.aspx.

Code behind business logic:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using AlumniPortal.BO;

namespace AlumniPortal
{
    public partial class Login : System.Web.UI.Page
    {

```

```

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        BO.SessionManager.SessionInfo.student = null;
    }
}

protected void btnRegister_Click(object sender, EventArgs e)
{
    try
    {
        StudntRegBO objstudentDetails = new StudntRegBO
        {
            studentId = txtstudentid.Value,
            student_Fname = fname.Value,
            student_Lname = lname.Value,
            gender = gender.Value,
            graduate_year = graduateYear.Value,
            email = email.Value,
            phone = phone.Value,
            address = address.Value,
            username = username.Value,
            password = password.Value,
            role = "student",
            Company = company.Value
        };
        DAL objDal = new DAL();
        int result = objDal.studentRegistration(objstudentDetails);
        ClientScript.RegisterStartupScript(GetType(), "alert", "alert('Regeistration Sucessfull, Kindly wait for Admin Approval');", true);
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

protected void btnSubmit_Click(object sender, EventArgs e)
{
    DAL objDal = new DAL();
    var userDetails = objDal.validateUser(Lusername.Value, Lpassword.Value);
    if (userDetails != null)
    {

```

```

BO.SessionManager.SessionInfo.student = userDetails;
if (BO.SessionManager.SessionInfo.student.role == "admin")
{
    Response.Redirect("Dashboard.aspx");
}
else if (BO.SessionManager.SessionInfo.student.role == "event")
{
    Response.Redirect("eventDashboard.aspx");
}
else if (BO.SessionManager.SessionInfo.student.role == "student")
{
    Response.Redirect("studDashboard.aspx");
}
}
else
{
    ClientScript.RegisterStartupScript(GetType(), "alert", "alert('Invalid Credentials');",
true);
}
}
}
}

```

Design view of Home.aspx:

The design view of Home.aspx shows a login form for the Governors State University Alumni Portal. The header features the university's logo and name. The main form contains two input fields for username and password, a blue 'LOG IN' button, and a 'Forgot Password' link. A red dashed box highlights the 'InvalidLbl' label, indicating an error message.

Figure 9(Design view of Home.aspx)

Dashboard.aspx: The code in the Dashboard.aspx mainly deals with the administrator functionalities.

Code behind business logic:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using AlumniPortal.BO;

namespace AlumniPortal
{
    public partial class Dashboard : System.Web.UI.Page
    {
        DAL objDal = new DAL();
        protected void Page_Load(object sender, EventArgs e)
        {
            Response.Cache.SetNoStore();
            if (BO.SessionManager.SessionInfo.student != null)
            {
                if (!IsPostBack)
                {
                    gvbind();
                }
            }
            else
            {
                Response.Redirect("Login.aspx");
            }
        }
        public void gvbind()
        {
            grdDetails.DataSource = objDal.getStudentDetails("");
            grdDetails.DataBind();
            grdView.DataSource = objDal.geteventDetails();
            grdView.DataBind();
        }

        protected void grdDetails_RowUpdating(object sender, GridViewCommandEventArgs e)
        {
            if (e.CommandName == "Approve")
            {
                int index = Convert.ToInt32(e.CommandArgument.ToString());
                string id = grdDetails.Rows[index].Cells[0].Text.ToString();
            }
        }
    }
}
```

```

        string name = grdDetails.Rows[index].Cells[1].Text.ToString();
        DAL objDal = new DAL();
        objDal.approveStudent(id.ToString(),name.ToString());
        gvbind();
    }
}
protected void gridView_RowEditing(object sender, GridViewEditEventArgs e)
{
    gridView.EditIndex = e.NewEditIndex;
    gvbind();
}
protected void gridView_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    int userid = Convert.ToInt32(gridView.DataKeys[e.RowIndex].Value.ToString());
    GridViewRow row = (GridViewRow)gridView.Rows[e.RowIndex];

    eventBo objeventBo = new eventBo {
        eventId = userid,
        name = ((TextBox)row.Cells[1].Controls[0]).Text,
        date = ((TextBox)row.Cells[2].Controls[0]).Text.ToString(),
        time = ((TextBox)row.Cells[3].Controls[0]).Text.ToString(),
        description = ((TextBox)row.Cells[4].Controls[0]).Text.ToString(),
        venue = ((TextBox)row.Cells[5].Controls[0]).Text.ToString(),
        number = ((TextBox)row.Cells[6].Controls[0]).Text.ToString()
    };
    objDal.updateevent(objeventBo);
    gvbind();
}
protected void gridView_RowCancelingEdit(object sender, GridViewCancelEventArgs
e)
{
    gridView.EditIndex = -1;
    gvbind();
}
}

```

Design View:

The design view of Dashboard.aspx consists of two main sections: 'Student Details' and 'Events'.

**Student Details**

Student Id	First Name	Last Name	Gender	Email	Phone	Address	Is approved	
Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Approve/Reject
Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Approve/Reject
Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Approve/Reject
Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Approve/Reject
Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Approve/Reject

**Events**

Event Id	Event Name	Date	Time	Description	Venue	Ph. No	
Databound	Databound	Databound	Databound	Databound	Databound	Databound	Edit
Databound	Databound	Databound	Databound	Databound	Databound	Databound	Edit
Databound	Databound	Databound	Databound	Databound	Databound	Databound	Edit
Databound	Databound	Databound	Databound	Databound	Databound	Databound	Edit
Databound	Databound	Databound	Databound	Databound	Databound	Databound	Edit

Figure 10(Design view of Dashboard.aspx)

Eventdashboard.aspx: The code in eventdashboard.aspx mainly illustrates the Event manager functionalities.

Code behind Business logic:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using AlumniPortal.BO;
using System.Data;
using Twilio;

namespace AlumniPortal
{
    public partial class eventDashboard : System.Web.UI.Page
    {
        DAL objDal = new DAL();
        protected void Page_Load(object sender, EventArgs e)
        {
            Response.Cache.SetNoStore();
            if (BO.SessionManager.SessionInfo.student != null)
```

```

    {
        if (!IsPostBack)
        {
            gvbind();
        }
    }
    else
    {
        Response.Redirect("Login.aspx");
    }
}
public void gvbind()
{
    grdView.DataSource = objDal.geteventDetails();
    grdView.DataBind();
}

protected void grdView_RowEditing(object sender, GridViewEditEventArgs e)
{
    grdView.EditIndex = e.NewEditIndex;
    gvbind();
}
protected void grdView_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    int userid = Convert.ToInt32(grdView.DataKeys[e.RowIndex].Value.ToString());
    GridViewRow row = (GridViewRow)grdView.Rows[e.RowIndex];

    eventBo objeventBo = new eventBo {
        eventId=userid,
        name = ((TextBox)row.Cells[0].Controls[0]).Text,
        date = ((TextBox)row.Cells[1].Controls[0]).Text.ToString(),
        time = ((TextBox)row.Cells[2].Controls[0]).Text.ToString(),
        description = ((TextBox)row.Cells[3].Controls[0]).Text.ToString(),
        venue = ((TextBox)row.Cells[4].Controls[0]).Text.ToString(),
        number = ((TextBox)row.Cells[5].Controls[0]).Text.ToString()
    };
    objDal.updateevent(objeventBo);
    gvbind();
}
protected void grdView_RowCancelingEdit(object sender, GridViewCancelEventArgs e)
{
    grdView.EditIndex = -1;
    gvbind();
}

```



```

}

protected void Calendar1_SelectionChanged(object sender, EventArgs e)
{
    string dateselectd = Calendar1.SelectedDate.Date.ToString("MM/dd/yyyy");
    string eventDescription="No Eevnts";
    string eventdate="";
    eventdesbox.Text = "";
    DataSet ds = new DataSet();
    ds = objDal.geteventDetails();
    for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
    {
        eventDescription = ds.Tables[0].Rows[i]["description"].ToString();
        eventdate = ds.Tables[0].Rows[i]["date"].ToString();
        if (dateselectd == eventdate)
        {
            eventdesbox.Text = eventDescription.ToString();
        }
    }
    if(eventdesbox.Text.Trim()=="")
    {
        eventdesbox.Text = "Whoops! No events found";
    }
}

protected void Button1_Click(object sender, EventArgs e)
{
    try
    {
        // Find your Account Sid and Auth Token at twilio.com/user/account
        string AccountSid = "AC308c41d8ca8b60924ea3deb2e3489e69";
        string AuthToken = "85bacc582d60e582757e2433ea8f08c1";
        string Number = "";

        DAL objDal = new DAL();
        DataTable ds = new DataTable();
        ds = objDal.getStudentDetails("").Tables[0];

        for (int i = 0; i < ds.Rows.Count; i++)
        {
            Number = ds.Rows[i]["phone"].ToString();

            TwilioRestClient client;
            client = new TwilioRestClient(AccountSid, AuthToken);
        }
    }
    catch { }
}

```

```
Message result = client.SendMessage("+16306353090", Number,
eventdesbox.Text);
```

```
    if (result.RestException != null)
    {
        //an exception occurred making the REST call
        string message = result.RestException.Message;
    }
}

catch (Exception)
{
    throw;
}
}
```

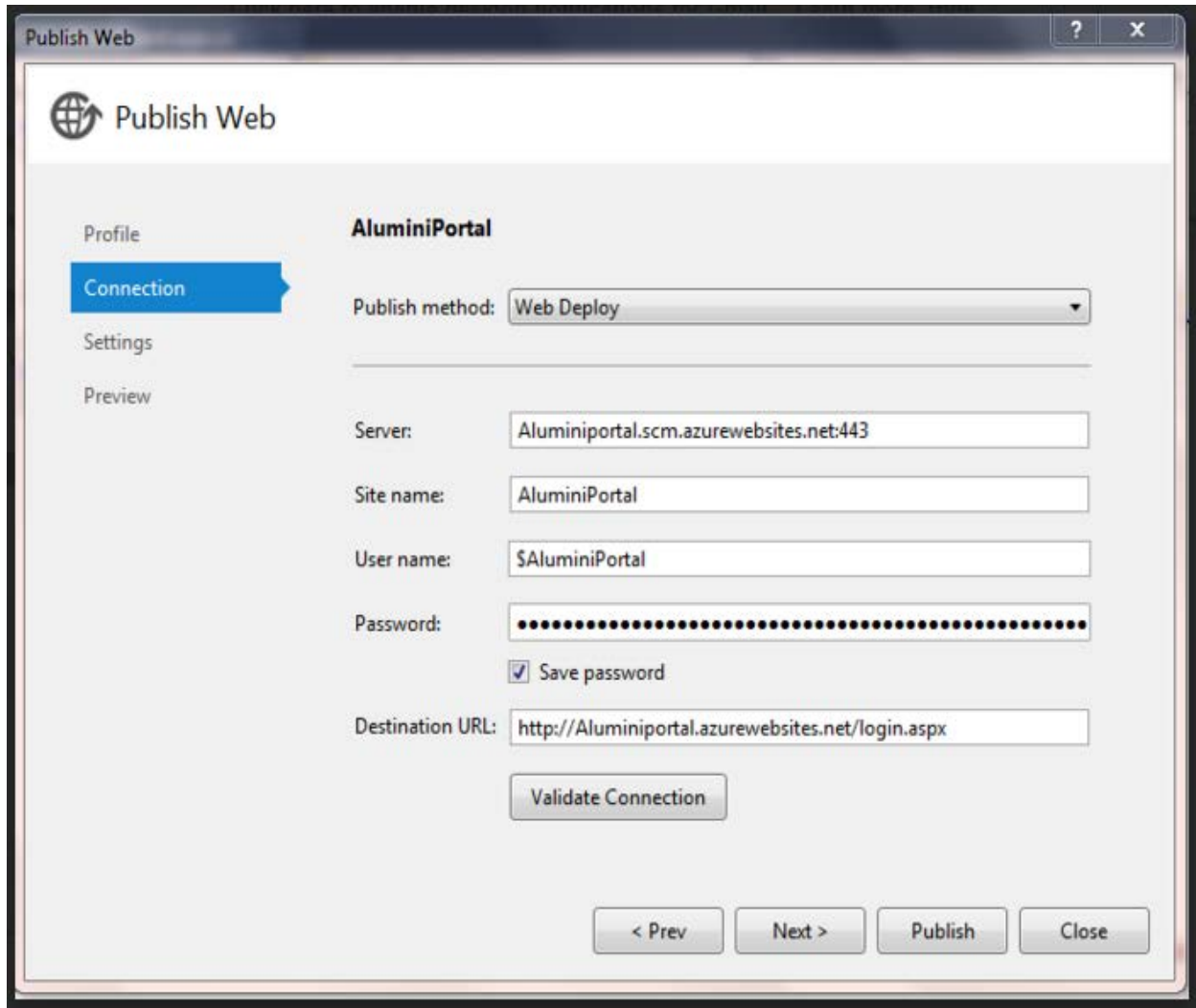
Design View:

The design view of EventDashboard.aspx features a dark header bar with the text "Alumni Portal" on the left and a hamburger menu icon on the right. Below the header, the main content area has a light blue background. It starts with a section titled "Events" followed by a table with 7 columns: Event Name, Date, Time, Description, Venue, Ph. No, and an Edit link. Each row contains "Databound" text for the first six columns and an "Edit" link. Below this is a section titled "Select Date of events" which contains a calendar widget. The calendar shows three months: October, November, and December. The days of the week are listed as Su, Mo, Tu, We, Th, Fr, Sa. The dates 25, 26, 27, 28, 29, 30, and 31 are visible for October, and 1, 2, 3, 4, 5, 6, and 7 are visible for November. To the right of the calendar is a large, empty rectangular area with a vertical scrollbar.

Figure 11(Design view of EventDashboard.aspx)

## 5 *Project Design Units Impacts*

After successful azure sql database design and asp.net front end pages, by using webapps in windows azure both sql azure database and asp.net web pages are connected. Publish is profile is created in windows azure in order to establish a connection between azure sql databases and windows azure web apps. For proper functioning of Gsu alumni portal web app publishing the webapp from visual studio is important.



The image shows the 'Publish Web' dialog box in Visual Studio. The title bar says 'Publish Web'. On the left, there is a sidebar with four tabs: 'Profile', 'Connection' (which is selected and highlighted in blue), 'Settings', and 'Preview'. The main area is titled 'AluminiPortal'. It contains the following fields and controls:

- Publish method:** A dropdown menu set to 'Web Deploy'.
- Server:** A text box containing 'Aluminiportal.scm.azurewebsites.net:443'.
- Site name:** A text box containing 'AluminiPortal'.
- User name:** A text box containing '\$AluminiPortal'.
- Password:** A text box filled with dots, with a checkbox labeled 'Save password' checked below it.
- Destination URL:** A text box containing 'http://Aluminiportal.azurewebsites.net/login.aspx'.
- Buttons:** A 'Validate Connection' button is located below the Destination URL. At the bottom of the dialog are four buttons: '< Prev', 'Next >', 'Publish', and 'Close'.

Figure 12(Publish profile)

When visual studio is connected to windows azure, the azure sdk automatically generates a publish profile.

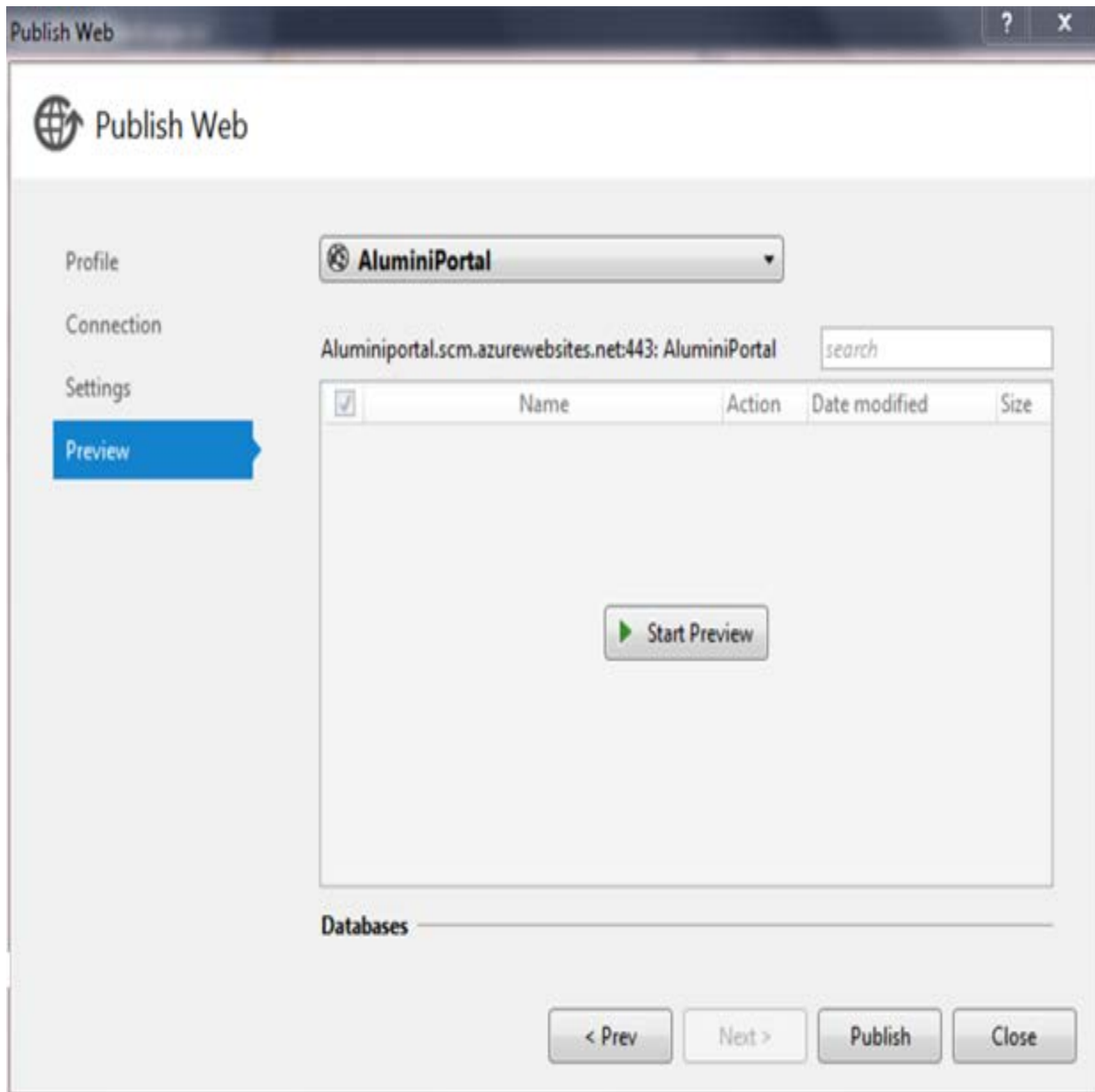
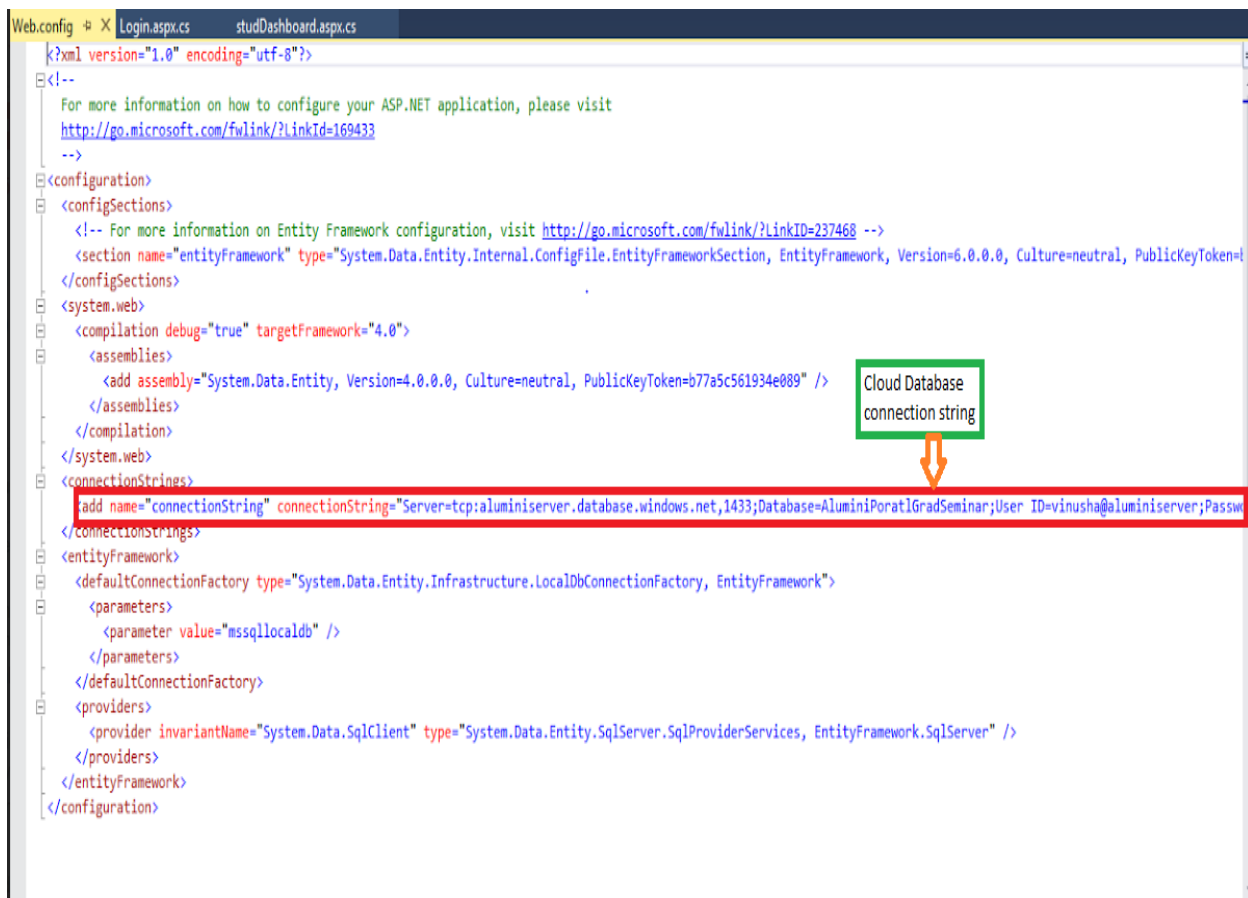
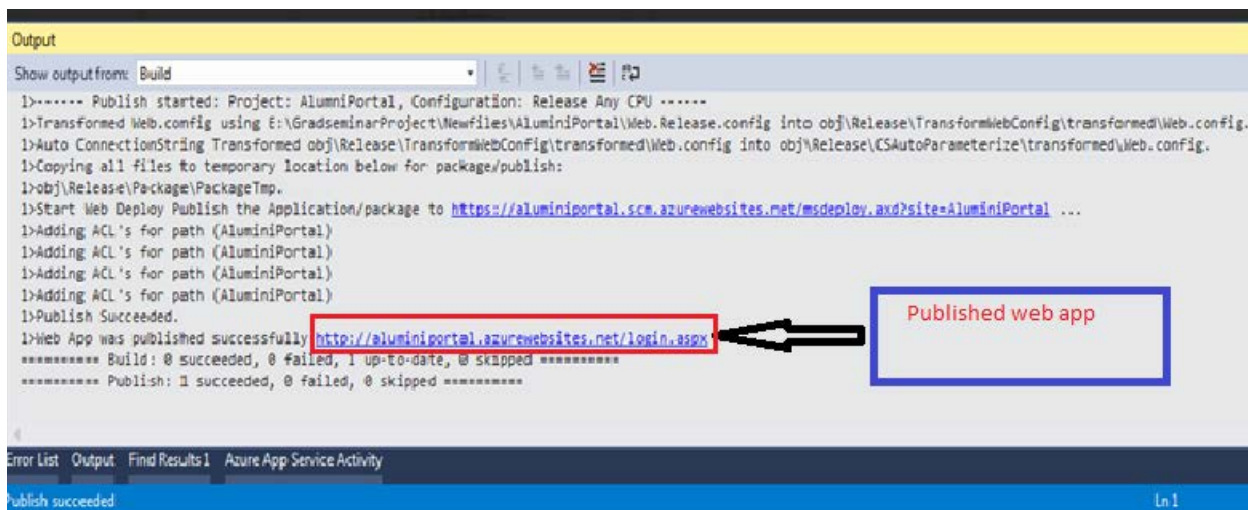


Figure 13(Publishing of Web app)

Azure sql database connection string is mentioned in web.config. During publishing the web app in cloud, the cloud database is connected instead of local databases.



```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkID=169433
-->
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=
  </configSections>
  <system.web>
    <compilation debug="true" targetFramework="4.0">
      <assemblies>
        <add assembly="System.Data.Entity, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
      </assemblies>
    </compilation>
  </system.web>
  <connectionStrings>
    <add name="connectionString" connectionString="Server=tcp:aluminiserver.database.windows.net,1433;Database=AluminiPortalGradSeminar;User ID=vinusha@aluminiserver;Passw
  </connectionStrings>
  <entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory, EntityFramework">
      <parameters>
        <parameter value="mssqllocaldb" />
      </parameters>
    </defaultConnectionFactory>
    <providers>
      <provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
    </providers>
  </entityFramework>
</configuration>
```



```
Output
Show output from: Build
1>----- Publish started: Project: AlumniPortal, Configuration: Release Any CPU -----
1>Transformed Web.config using E:\GradseminarProject\Newfiles\AluminiPortal\Web.Release.config into obj\Release\TransformWebConfig\transformed\Web.config.
1>Auto ConnectionString Transformed obj\Release\TransformWebConfig\transformed\Web.config into obj\Release\CSAutoParameterize\transformed\Web.config.
1>Copying all files to temporary location below for package/publish:
1>obj\Release\Package\PackageTmp.
1>Start Web Deploy Publish the Application/package to https://alumniportal.azurewebsites.net/msdeploy.axd?site=AluminiPortal ...
1>Adding ACL's for path (AluminiPortal)
1>Adding ACL's for path (AluminiPortal)
1>Adding ACL's for path (AluminiPortal)
1>Adding ACL's for path (AluminiPortal)
1>Publish Succeeded.
1>Web App was published successfully http://alumniportal.azurewebsites.net/login.aspx
***** Build: 0 succeeded, 0 failed, 1 up-to-date, 0 skipped *****
***** Publish: 1 succeeded, 0 failed, 0 skipped *****
```

The published GSU alumni portal web app destination URL is <http://alumniportal.azurewebsites.net/login.aspx>.

In microsoft azure portal we can view all the services used to host the web app in cloud.

Microsoft Azure | CREDIT STATUS | vvmuni@student.govst.edu

ALL ITEMS | all items

NAME	TYPE	STATUS	SUBSCRIPTL...	LOCAT... ↑
aluministorageacc	Storage Account	✓ Online	Azure Pass	Central US
AluminiPortal	Web app	✓ Running	Azure Pass	East US
AluminiPoratiGradSeminar	SQL Database	✓ Online	Azure Pass	East US
Governors State University	Directory	✓ Active	Shared by al...	United States

WEB APPS 1

VIRTUAL MACHINES 0

MOBILE SERVICES 0

CLOUD SERVICES 0

BATCH SERVICES 0

SQL DATABASES 1

STORAGE 1

HIDINGSIGHT 0

MEDIA SERVICES 0

SERVICE BUS 0

Web app

sql Database

Figure 14(Microsoft azure Services used to host a web application)

The usage overview of Gsu alumni portal web app.

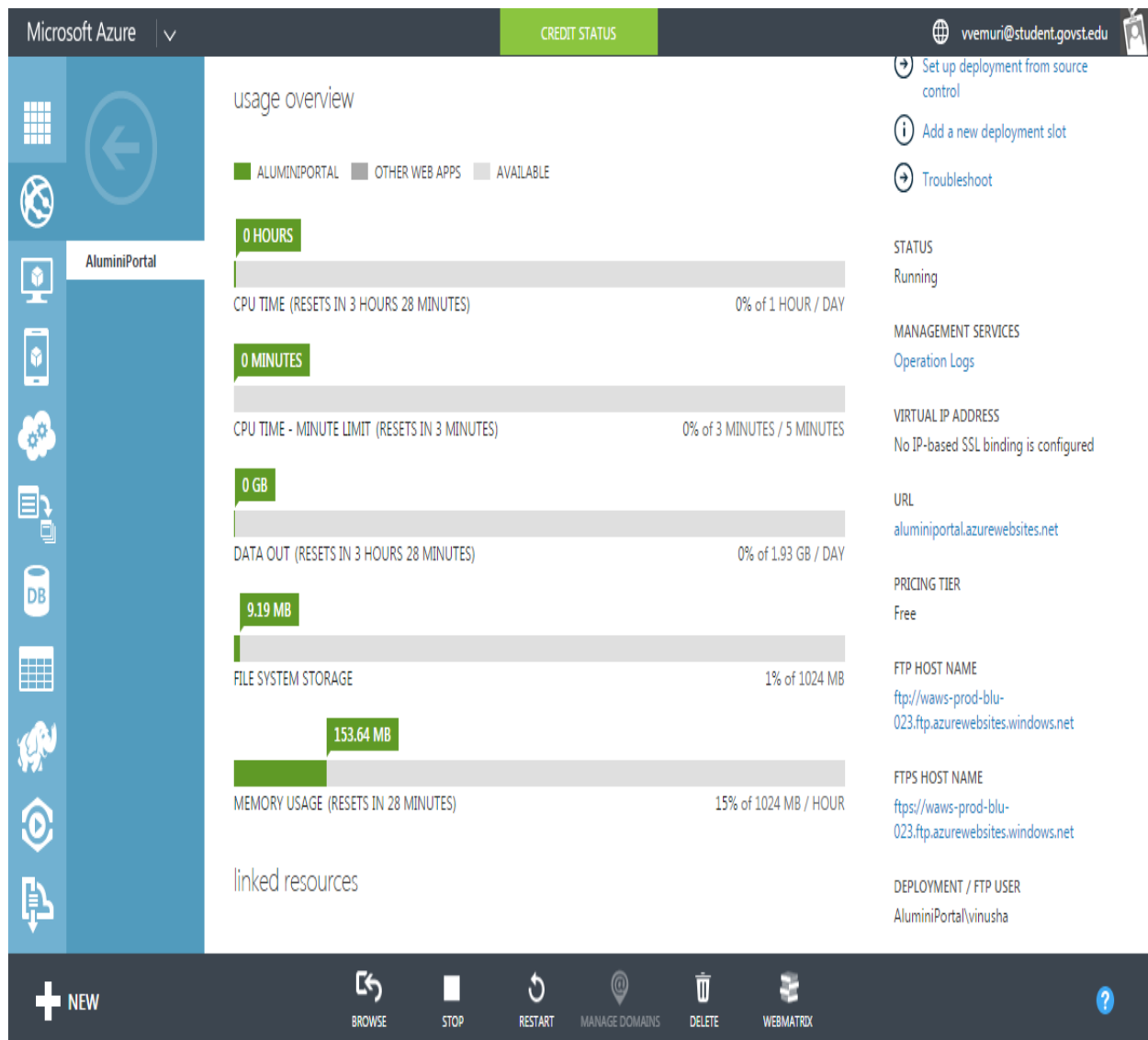


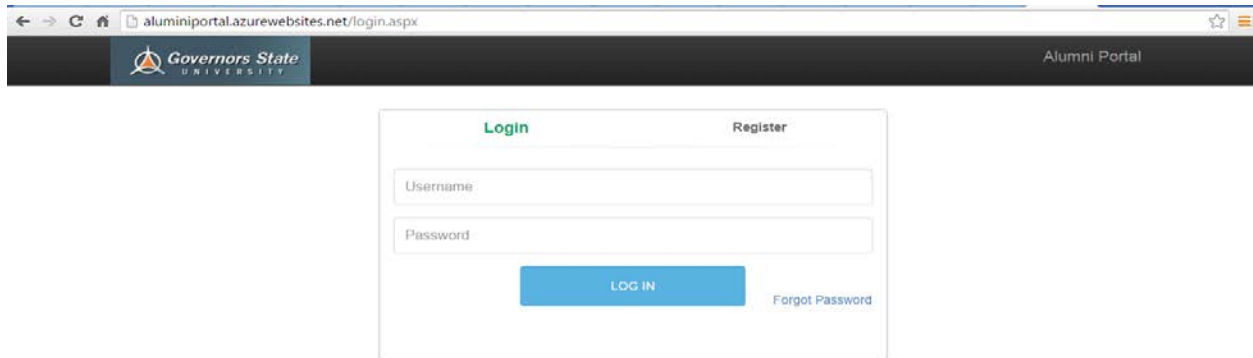
Figure 15(Usage Overview)

## 5.1 Functional Area/Design Unit A

### 5.1.1 Functional Overview

After successful deployment, the Gsu alumni portal web app functional view is described as:

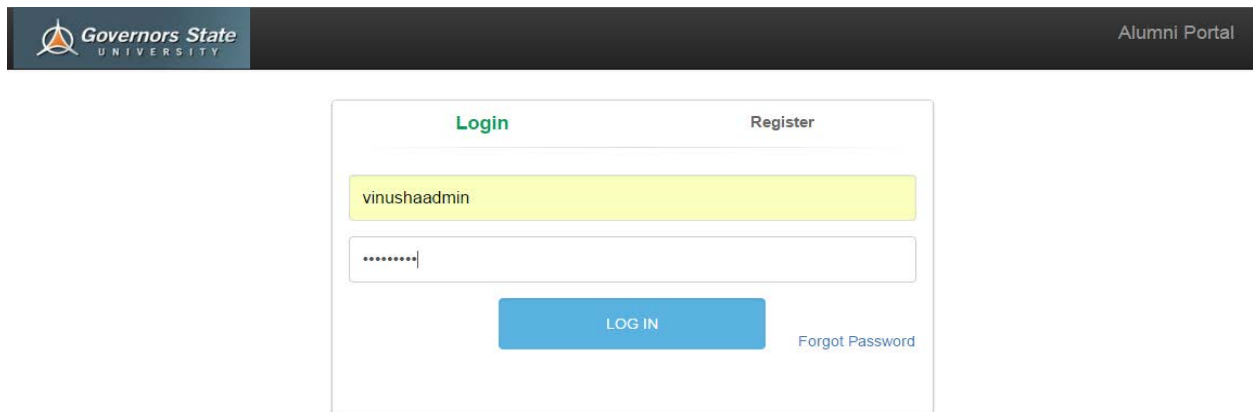
- Login: This general login web page for Administrator, Event manager and student.



The screenshot shows a web browser window with the URL `alumniportal.azurewebsites.net/login.aspx`. The header features the Governors State University logo and the text "Alumni Portal". The main content area contains a login form with two tabs: "Login" (active) and "Register". The login form has two input fields: "Username" and "Password". Below the password field is a blue "LOG IN" button and a link for "Forgot Password".

Figure 16(General login Page)

- Administrator: The Administrator logs in with his/her login user id and password.



The screenshot shows the same login form as Figure 16, but with the "Administrator" user logged in. The "Username" field is highlighted in yellow and contains the text "vinushaadmin". The "Password" field is filled with dots. The "LOG IN" button is still visible, and the "Forgot Password" link is also present.

Figure 17(Administrator login)

After successful administrator login, he/she can view the existing student, the student waiting for approval, list of events and he can add new employee (administrator or event Manager).



## Student Details

Student Id	First Name	Last Name	Gender	Email	Phone	Address	Is approved	
1152689	vinusha	vemuri	F	vinushavemuri@gmail.com	6302209682	schaumburg il	True	<a href="#">Approve/Reject</a>
admin	vinusha chowdary		F	vinusha.chowdary19@gamil.com	6302209682	hoffman estates	True	<a href="#">Approve/Reject</a>
event	vinusha paladugu		F	vinusha.chowdary19@gamil.com	6302209682	Chicago	True	<a href="#">Approve/Reject</a>
1152690	Naresh	palagudu	M	naresh.sears@gmail.com	2244654878	california	True	<a href="#">Approve/Reject</a>
786786	yashwanth	k	M	yashwanth.kalwa@gmail.com	8562208667	sample	True	<a href="#">Approve/Reject</a>
admin	soon-o-park		F	SPark@govst.edu	2244654878	GSU	True	<a href="#">Approve/Reject</a>
123456	pal	nar	M	naresh.sears@gmail.com	2244654878	1510 valley lake drive	True	<a href="#">Approve/Reject</a>
admin	sairam dasu		M	sairamthe92@gmail.com	7086794199	3324 craig dr	True	<a href="#">Approve/Reject</a>
event	Spark		F	SPark@govst.edu	6308051155	govst	True	<a href="#">Approve/Reject</a>
admin	sindhu M		F	sindhu@gmail.com	5715283505	Govst	False	<a href="#">Approve/Reject</a>
1155250	Suresh	Kurapati	M	sureshkurapati88@gmail.com	4847970622	illions	True	<a href="#">Approve/Reject</a>
1122222	evans	evans	M	evans2@gmail.com	7089692735	Govst	True	<a href="#">Approve/Reject</a>
1155158	ajay	akarapa	M	ajayakarapu@gmail.com	2295369278	Govst	False	<a href="#">Approve/Reject</a>
1152688	Pruthvi	Prodduturi	M	Pruthvi.prodduturi@gmail.com	6302209682	abc	True	<a href="#">Approve/Reject</a>
event	Naresh Paladugu		M	naresh.sears@gmail.com	2244	illinois	False	<a href="#">Approve/Reject</a>
1153977	Mahesh babu	Chellu	M	Maheshbabuchellu@gmail	6302676167	Naperville	True	<a href="#">Approve/Reject</a>
event	Naresh Paladugu		M	naresh.sears@gmail.com	2244654878	schaumburg	True	<a href="#">Approve/Reject</a>
1122334455	John	Miller	M	john2@gmail.com	22441822	ohio	True	<a href="#">Approve/Reject</a>
121314	sam	samuel	M	sam@gmail.com	22443567	abc	False	<a href="#">Approve/Reject</a>

## Events

Event Id	Event Name	Date	Time	Description	Venue	Ph. No	
5	Gradseminar	11/23/2015	10:30 am	Final Presentation-11/23-10:30am	Gsu1210	6302209682	<a href="#">Edit</a>
6	Database design	11/23/2015	4:30Pm	Database design-final project presentation-11/23/2015-4:30pm	Cpsc-2340	6302209682	<a href="#">Edit</a>
7	testing	11/20/2015	3:30	Code testing	GSU	8562208667	<a href="#">Edit</a>
8	Documentation	11/30/2015	7:30 Pm	All batches should attend	Seminar Hall	6302209682	<a href="#">Edit</a>

Figure 18(Administrator can approve or reject the student registrations)

a) Add Employee:

Alumni Portal
Home
Add Employee
Logout

Username:

Name:

Phone Number:

email:

password:

Gender:

Address:

UserType:  
Admin

Save

© Copyright 2015 Vinusha V

Figure 19(Administrator adds an employee)

- Event Manager: The Event Manager Logins with his/her user id and password, event manger can add new event and send event notification sms to all registered users in the Gsu alumni portal.

Alumni Portal
Home
Add Event
Logout

### Events

Event Name	Date	Time	Description	Venue	Ph. No	
Gradseminar	11/23/2015	10:30 am	Final Presentation-11/23-10:30am	Gsu1210	6302209682	<a href="#">Edit</a>
Database design	11/23/2015	4:30Pm	Database design-final project presentation-11/23/2015-4:30pm	Cpsc-2340	6302209682	<a href="#">Edit</a>
testing	11/20/2015	3:30	Code testing	GSU	8562208667	<a href="#">Edit</a>
Documentation	11/30/2015	7:30 Pm	All batches should attend	Seminar Hall	6302209682	<a href="#">Edit</a>
Project Documentation	11/25/2015	10:30 Am	Project Documentation (Seminar hall) - 11/25/2015 10:30 am	Seminar Hall	6302209682	<a href="#">Edit</a>

### Select Date of events

October
November
December

Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Project Documentation (Seminar hall) - 11/25/2015 10:30 am

SEND SMS

Figure 20(Event Manager sends an event sms)

a) Add Event:

The screenshot shows the 'Alumni Portal' header in a dark bar. Below it, the title 'Add Event' is centered. The form is divided into two columns. The left column contains fields for 'Name:', 'Time:', and 'Phone Number:'. The right column contains fields for 'Date:', 'Venue:', and 'Description:'. A 'Save' button is located at the bottom left of the form. At the bottom of the page, there is a copyright notice: '© Copyright 2015 Vinusha V'.

Figure 21(Event manager adds an event)

- Student: After successful registration and acceptance from administrator, student can login into Gsu alumni portal and he/she can view the contact information about other registered students, list of events conducted by the university, Email other students and ask queries and can get answers from the administrator.

Student login:

The screenshot shows the 'Alumni Portal' header in a dark bar. Below it, the 'Governors State UNIVERSITY' logo is on the left. The main content area contains a registration form with two tabs: 'Login' and 'Register' (which is highlighted in green). The form fields include: 'Username', 'Email Address', 'Password', 'Confirm Password', 'Student Id', 'First Name', 'Last Name', a gender dropdown menu (currently showing 'M'), and 'Graduate Year'.

Figure 22(Student Registration)

Login

Register

LOG IN

Forgot Password

Figure 23(Student Login)

## Student Details

Student Id	First Name	Last Name	Gender	Email	Phone
1152689	vinusha	vemuri	F	vinushavemuri@gmail.com	6302209682
1152690	Naresh	palagudu	M	naresh.sears@gmail.com	2244654878
786786	yashwanth	k	M	yashwanth.kalwa@gmail.com	8562208667
123456	pal	nar	M	naresh.sears@gmail.com	2244654878
1155250	Suresh	Kurapati	M	sureshkurapati88@gmail.com	4847970622
1122222	evans	evans	M	evans2@gmail.com	7089692735
1152688	Pruthvi	Prodduturi	M	Pruthvi.prodduturi@gmail.com	6302209682
1153977	Mahesh babu	Chellu	M	Maheshbabuchellu@gmail	6302676167
1122334455	John	Miller	M	john2@gmail.com	22441822

## Events

Event Id	Event Name	Date	Time	Description	Venue	Ph. No
5	Gradseminar	11/23/2015	10:30 am	Final Presentation-11/23-10:30am	Gsu1210	6302209682
6	Database design	11/23/2015	4:30Pm	Database design-final project presentation-11/23/2015-4:30pm	Cpsc-2340	6302209682
7	testing	11/20/2015	3:30	Code testing	GSU	8562208667
8	Documentation	11/20/2015	7:30	All batches should attend	Seminar	6302209682

Figure 24(Gsu alumni portal student view)

- Edit Profile:

Alumni Portal

Home Mail Ask **Edit Profile** Logout

**Username:**  
vinusha

**First Name:**  
vinusha

**Gender:**  
F

**Gender:**  
2015

**Address:**  
schaumburg il

**Password:**

**Last Name:**  
vemuri

**Student ID:**  
1152689

**Phone Number:**  
6302209682

**email:**  
vinushavemuri@gmail.com

Save

© Copyright 2015 Vinusha V

Figure 25(Edit Profile)

- Ask admin:

Alumni Portal

Home Mail **Ask** Edit Profile Logout

**Query:**

**Description:**

Ask

© Copyright 2015 Vinusha V

Figure 26(Sends queries to admin)

- Email:

The screenshot shows the 'Alumni Portal' header with navigation links: Home, Mail (highlighted), Ask, Edit Profile, and Logout. Below the header is a form for sending an email. It includes three input fields: 'To:', 'Subject:', and 'Body:'. The 'Body:' field is a larger text area. At the bottom of the form is a 'Send Mail' button.

Alumni Portal

Home Mail Ask Edit Profile Logout

To:

Subject:

Body:

Send Mail

Figure 27(Email other students)

## **6    *Conclusion***

While developing Gsu alumni portal, we have gathered immense knowledge about Asp.net technology and cloud computing. We have understood how a web application works and how can it be made as a user friendly application. It also provides knowledge about the latest technology used in developing applications and client server technology that will have great demand in future. This will provide us better opportunities and guidance in future in developing projects independently.

## **7 Acknowledgements**

It gives me immense pleasure to express my deep sense of gratitude to Professor **Dr. Soon-Ok Park**, who has been kind enough to guide us in the planning the GSU alumni project. We are very thankful to her for helping us in all possible ways during this project work in field of Computer Science, at Governors State University.

I am also thankful for her continuous feedback and encouragement throughout this project work. Her expertise in various subjects helped me to gain knowledge in various aspects, which will be a great help in my future endeavours.

Finally, I would like to extend thanks to my team members for their continuous support, discussions and suggestions in developing the Gsu alumni project.



## 8 *References*

- For installing visual studio 2015

<https://msdn.microsoft.com/en-us/library/e2h7fzkw.aspx>

- For installing Sql server 2014

<https://msdn.microsoft.com/en-us/library/ms143219.aspx>

- For creating Asp.net web pages

<https://msdn.microsoft.com/en-us/library/fddycb06.aspx>

- To create database in sql azure

<https://azure.microsoft.com/en-us/documentation/articles/sql-database-get-started/>

- To create web app in windows azure

<https://azure.microsoft.com/en-us/documentation/articles/web-sites-dotnet-get-started/>

- To publish web app using visual studio 2015

[http://blogs.msdn.com/b/uk\\_faculty\\_connection/archive/2015/07/19/using-visual-studio-2015-and-deploying-your-first-web-app-to-your-free-azure-subscription.aspx](http://blogs.msdn.com/b/uk_faculty_connection/archive/2015/07/19/using-visual-studio-2015-and-deploying-your-first-web-app-to-your-free-azure-subscription.aspx)

***List of Figures:***

- Figure 1 - Asp.net web application architecture
- Figure 2 - 3- Tier Azure Deployment
- Figure 3 - Sequence Diagram
- Figure 4 - Azure sql database login
- Figure 5 - Azure sql database name
- Figure 6 - Database parameters(list of tables)
- Figure 7 - List of stored procedures
- Figure 8 - Dependencies
- Figure 9 - Design view of home.aspx
- Figure 10 - Design view of Dashboard.aspx
- Figure 11 - Design view of Eventdashboard.aspx
- Figure 12 - Publishing profile
- Figure 13 - Publishing of web app
- Figure 14 - Microsoft azure Services used to host a web application
- Figure 15 - Usage overview
- Figure 16 - General login page
- Figure 17 - Administrator login
- Figure 18 - Administrator can approve or reject the student registrations
- Figure 19 - Administrator adds an employee
- Figure 20 - Event Manager sends an event sms to the registered users in the portal
- Figure 21 - Event manager adds an event
- Figure 22 - Student Registration
- Figure 23 - Student Login
- Figure 24 - Gsu alumni portal student view
- Figure 25 - Edit Profile
- Figure 26 - Sends queries to admin
- Figure 27 - Email other students